International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

DESIGNING AN ANALYTICAL TOOL FOR TEMPERATURE FORECAST FROM THE SOLAR RENEWABLE ENERGY COMPUTATION

G.KAVITHA [1], B.GOPALAKRISHNAN [2], K.V.DUVARAKESH [3], B.AVINASH REDDY [4]

[1] Assistant Professor, Department of Information Technology,
Prathyusha Institute of Technology and Management,
Poonamalee - Thiruvallur road, Thiruvallur - 602025, Tamilnadu, India
kavithaaol@yahoo.com

[2] Final Year Student, Department of Information Technology,
Prathyusha Institute of Technology and Management,
Poonamalee - Thiruvallur road, Thiruvallur - 602025, Tamilnadu, India
gol.d.gokrish@gmail.com

[3] Final Year Student, Department of Information Technology,
Prathyusha Institute of Technology and Management,
Poonamalee - Thiruvallur road, Thiruvallur - 602025, Tamilnadu, India
kvduvarakesh@gmail.com

[4] Final Year Student, Department of Information Technology,
Prathyusha Institute of Technology and Management,
Poonamalee - Thiruvallur road, Thiruvallur - 602025, Tamilnadu, India
avinashreddy.boddu@gmail.com

Abstract: **Being able to accurately predict temperature from the solar power hitting the photovoltaic panels is a key to estimate the various parameters which helps in temperature prediction. The solar power coming to our planet is predictable, but the energy produced fluctuates with varying atmospheric conditions. Usually, numerical weather prediction models are used to make irradiation forecasts. Our study based on back propagation neural network which is trained and tested based on dataset provided. This paper utilizes artificial neural networks for temperature forecasting. Our study based on back propagation neural network which is trained and tested based on dataset provided. In formulating the ANN-based predictive model; three-layer network has been constructed. Suitable air temperature predictions can provide farmers and producers with valuable information when they face decisions regarding the use of mitigating technologies such as orchard heaters or irrigation. Temperature warnings are important forecasts because they are used to protect life and property. Temperature forecasting is the application of science and technology to predict the state of the temperature for a future time and a given location. Temperature forecasts are made by collecting quantitative data about the current state of the atmosphere. In this paper, a neural network-based algorithm for predicting the temperature is presented. The Neural Networks package supports different types of training or learning algorithms. One such algorithm is Back Propagation Neural Network (BPN) technique. The main advantage of the BPN neural network method is that it can fairly approximate a large class of functions. This method is more efficient than numerical differentiation. The simple meaning of this term is that our model has potential to capture the complex relationships between many factors that contribute to certain temperature. The proposed idea is tested using the real time dataset. The results are compared with practical working of meteorological department and these results confirm that our model have the potential for successful application to temperature forecasting. Real time processing of weather data indicate that the BPN based weather forecast have shown improvement not only over guidance forecasts from numerical models, but over official local weather service forecasts as well. Artificial neural networks and the back propagation algorithm used for temperature forecasting in general are explained.**

Keywords: Solar Energy Computation, Temperature Prediction, Back Propagation, Artificial Neural Network

1. Introduction

Temperature prediction is a temporal and time series based process. Accurate forecasting is important in today's world as agricultural and industrial sectors are largely dependent on the temperature. Due to non-linearity in climatic physics, neural networks are suitable to predict these meteorological processes. Back propagation integrated with genetic algorithm is the
most important algorithm to train neural networks. In this paper, in order to show the dependence of temperature on a particular data series, a time series based temperature

prediction model using integrated back propagation with genetic algorithm technique is proposed. In the proposed technique, the effect of under training and over training the system is also shown. The test results of the technique are enlisted along with back propagation and Neural Network.

Being able to accurately predict the temperature with the help of solar rays hitting the photovoltaic panels is a key challenge to integrate more and more renewable energy sources into the grid, as the total power generation needs to match the predicted temperature value. The solar power coming to our planet is predictable, but the energy produced fluctuates with varying atmospheric conditions. By using this property, Temperature prediction can be done. Usually, numerical weather prediction models are used to make irradiation forecasts. This project focuses on machine learning techniques to produce more accurate predictions for temperature using solar power generated.

Our strategy to make this prediction is:
collect, understand and process the weather data,

- perform different machine learning techniques to make the prediction,
- perform some feature engineering aside of the forecast features,
- analyze the results and predict the temperature value.

2. Methodology

A neural network [1] is a powerful data modeling tool that is able to capture and represent complex input /output relationships. The motivation for the development of neural network technology stemmed from the desire to implement an artificial system that could perform intelligent tasks similar to those performed by the human brain. Neural network resemble the human brain in the following two ways:

- ➢ A neural network acquires knowledge through learning.
- ➢ A neural network"s knowledge is stored within interneuron connection strengths known as synaptic weights

The true power and advantages of neural networks lies in the ability to represent both linear and non linear relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes for true modeling data that contains non linear characteristics.

A neural network model is a structure that can be adjusted to produce a mapping from a given set of data to features of or relationships among the data. The model is adjusted, or trained, using a collection of data from a given source as input, typically referred to as the training set. After successful training, the neural network will be capable to perform classification, estimation, prediction, or simulation on new data from the same or similar sources.

An Artificial Neural Network (ANN) [5] is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the new structure of the information processing system. It is composed of a huge number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a particular application, such as pattern recognition or data classification, through a learning process. Learning in biological systems adds adjustments to the synaptic connections that exist between the neurons.

A back propagation network [9] consists of at least three layers (multi layer perception): an input layer, at least one intermediate hidden layer, and an output layer. In contrast to the Interactive Activation and Competition (IAC) neural networks (IAC) and Hopfield networks, connection weights in a back propagation network are one way. Typically, input units are connected in a feed-forward fashion with input units fully connected to units in the hidden layer and hidden units fully connected to units in the output layer. An input pattern is propagated forward to the output units all the way through the intervening input-to-hidden and hidden-to-output weights when a Back Propagation network is cycled.

As the algorithm's name gives a meaning, the errors (and therefore the learning) propagate backwards from the output nodes to the inner nodes. So technically it can be explained, back propagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. This gradient is always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Frequently the term "back propagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent. Back propagation regularly allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited.

The proposed Temperature Prediction System using BPN Neural Network is tested using the dataset from [17]. The results are compared with practical temperature prediction results [18, 19]. This system helps the meteorologist to predict the future weather easily and accurately.

3. Related Work

Many works were done related to the temperature prediction system and BPN network. They are summarized below.

Y.Radhika and M.Shashi [3] presents an application of Support Vector Machines (SVMs) for weather prediction. Time series data of daily maximum temperature at location is studied to predict the maximum temperature of the next day at that location based on the daily maximum temperatures for a span of previous n days referred to as order of the input. Performance of the system is observed for various spans of 2 to 10 days by using optimal values of the kernel

Mohsen Hayati *et.al*, [5] studied about Artificial Neural Network based on MLP was trained and tested using ten years (1996-2006) meteorological data. The results show that

IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

MLP network has the minimum forecasting error and can be considered as a good method to model the short-term temperature forecasting [STTF] systems. Brian A. Smith *et.al*,[6] focused on developing ANN models with reduced average prediction error by increasing the number of distinct observations used in training, adding additional input terms that describe the date of an observation, increasing the duration of prior weather data included in each observation, and reexamining the number of hidden nodes used in the network. Models were created to forecast air temperature at hourly intervals from one to 12 hours ahead. Each ANN model, having a network architecture and set of associated parameters, was evaluated by instantiating and training 30 networks and calculating the mean absolute error (MAE) of the resulting networks for some set of input patterns.

Arvind Sharma *et.al*, [7] briefly explains how the different connectionist paradigms could be formulated using different learning methods and then investigates whether they can provide the required level of performance, which are sufficiently good and robust so as to provide a reliable forecast model for stock market indices. Experiment results exposes that all the connectionist paradigms considered could represent the stock indices behavior very accurately.

Mike O'Neill [11] focus on two major practical considerations: the relationship between the amounts of training data and error rate (corresponding to the effort to collect training data to build a model with given maximum error rate) and the transferability of models‟ expertise between different datasets (corresponding to the usefulness for general handwritten digit recognition).Henry A. Rowley eliminates the difficult task of manually selecting nonface training examples, which must be chosen to span the entire space of nonface images. Simple heuristics, like using the fact that faces rarely overlap in images, can further improve the accuracy. Comparisons with more than a few other state-of-the-art face detection systems are presented; showing that our system has comparable performance in terms of detection and false-positive rates..

4. ANN Approach

A. Phases in Backpropagation Technique

The back propagation [10] learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input is given through the neural network in order to generate the propagation's output activations.
2. Back propagation of the output activations propagation through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight Update

For each weight-synapse:

1. Multiply its input activation and output delta to get the gradient of the weight.
2. Bring the weight in the direction of the gradient by adding a ratio of it from the weight.

This ratio impacts on the speed and quality of learning; it is called the learning rate. The sign of the gradient of a weight designates where the error is increasing; this is why the weight must be updated in the opposite direction.

The phase 1 and 2 is repeated until the performance of the network is satisfactory.

B. Modes of Learning

There are essentially two modes of learning to choose from, one is on-line learning and the other is batch learning. Each propagation is followed immediately by a weight update in online learning [21]. In batch learning, much propagation occurs before weight updating occurs. Batch learning needs more memory capacity, but on-line learning requires more updates.

C. Algorithm

Actual algorithm for a 3-layer network (only one hidden layer):

1. Initialize the weights in the network (often randomly)
2. Do
3. For each example e in the training set
O = neural-net-output (network, e) ; forward pass
T = teacher output for e
4. Calculate error (T - O) at the output units
5. Compute delta_wh for all weights from hidden layer to output layer ; backward pass
6. Compute delta_wi for all weights from input layer to hidden layer ; backward pass continued
7. Update the weights in the network
8. Until all examples classified correctly or stopping criterion satisfied
9. Return the network

D. Back Propagation Neural Network

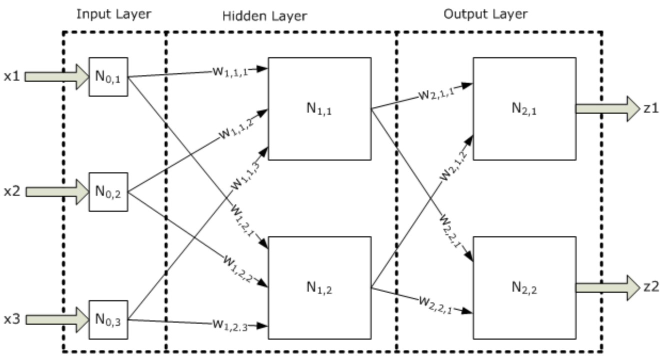

Fig1: A Back Propagation Neural Network Architecture

In the fig.1,

1. The output of a neuron in a layer moves to all neurons in the following layer.
2. Each neuron has its own input weights.
3. The weights for the input layer are assumed (fixed) to be 1 for each input. In other words, input values are not changed.
4. The output of the NN is obtained by applying input values to the input layer, passing the output of each neuron to the following layer as input.
5. The Back Propagation NN should have at least an input layer and an output layer. It could have zero or more hidden layers.

The number of neurons in the input layer is decided by the available number of possible inputs. The number of neurons in the output layer depends on the number of desired outputs. The number of hidden layers and number of neurons in each hidden layer cannot be well defined in advance, and could change per network configuration and type of data. Generally the addition of a hidden layer could allow the network to learn more complex patterns, but at the same time decreases its performance. A network configuration can have a single hidden layer, but more hidden layers can be added if the network is not learning well.

The following conditions are to be analyzed for input to BPN,

- Atmospheric Pressure
- Atmospheric Temperature
- Relative Humidty
- Wind Velocity and
- Wind Direction

Back propagation is an iterative process that starts with the last layer and moves backwards through the layers until the first layer is reached. Assume that for each layer, the error in the output of the layer is known. If the error of the output is known, then it is not hard to calculate changes for the weights, so as to reduce that error. The problem is that the error in the output of the very last layer only can be observed.

5. Training and Testing Neural Network

The best training procedure is to compile a wide range of examples (for more complex problems, more examples are required), which exhibit all the different characteristics of the problem. To create a robust and reliable network, in some cases, some noise or other randomness is added to the training data to get the network familiarized with noise and natural variability in real data . Poor training data inevitably leads to an unreliable and unpredictable network. Usually, the network is trained for a prefixed number of epochs or when the output error decreases below a particular error threshold. Special care is to be taken not to over train the network. By overtraining, the network may become too adapted in learning the samples from the training set, and thus may be unable to accurately classify samples outside of the training set.[Fig. 2]

A. Choosing the Number of Neurons

The number of hidden neurons affects how well the network is able to separate the data. A large number of hidden neurons will ensure correct learning, and the network is able to correctly predict the data it has been trained on, but its performance on new data, its ability to generalize, is compromised [5]. With too few hidden neurons, the network may be unable to learn the relationships amongst the data and the error will fail to fall below an acceptable level. Thus, selection of the number of hidden neurons is a crucial decision.

B. Choosing the Initial Weights

The learning algorithm uses a steepest descent technique, which rolls straight downhill in weight space until the first valley is reached. This makes the choice of initial starting point in the multidimensional weight space critical. However, there are no recommended rules for this selection except trying several different starting weight values to see if the network results are improved.

C. Choosing the Learning Rate

Learning rate effectively controls the size of the step that is taken in multidimensional weight space when each weight is modified. If the selected learning rate is too large, then the local minimum may be overstepped constantly, resulting in oscillations and slow convergence to the lower error state[6]. If the learning rate is too low, the number of iterations required may be too large, resulting in slow performance.

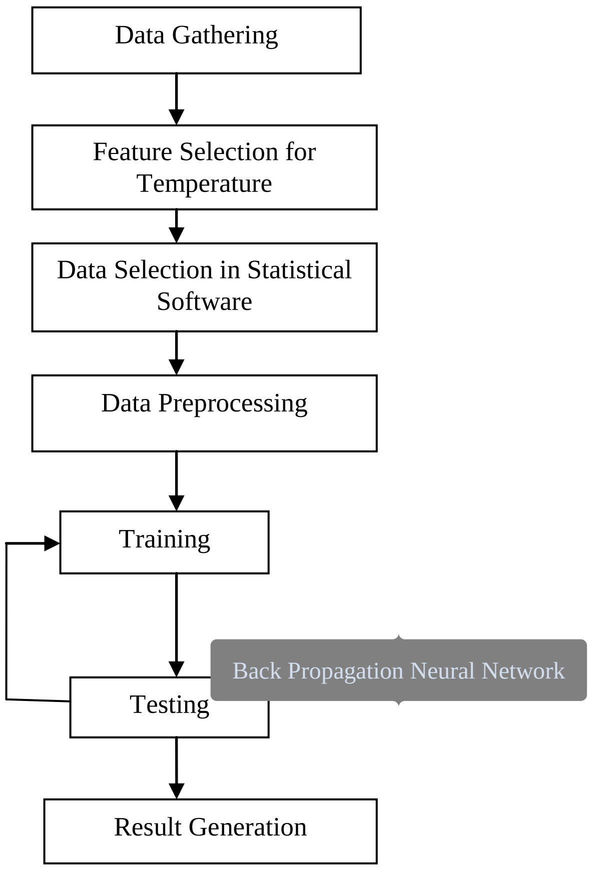

Figure 2: Flowchart of temperature prediction system

6. Result and Discussion

The obtained results indicate that satisfactory prediction accuracy has been achieved through back propagation neural network. A back propagation neural network with gradient descent method minimizes the error rate and it is a promising approach for temperature forecasting. Mean, minimum and maximum air temperatures were considered as input and output of the network.

Three different data sets were extracted from the input and target data for training, validation and test phases. While training set consists of 50 percent of data to build the model and determine the parameters such as weights and biases, validation data set includes 25 percent to measure the performance of network by holding constant parameters. Finally, 25 percent of data is used to increase the robustness of model in the test phase.

The validation and testing phases are very important due to misleading of small error in the training phase. If the network is not trained well due to the irrelevant data of the individual cases such as over fitting, it leads to the small error in the training set and makes large error during validation and test phases. While the purpose of training phase is based on learning, it is not a good metric for the performance of network in validation phase.

7. Conclusion

Neural-networks-based ensemble models were developed and applied for hourly temperature forecasting. The experimental results show that the ensemble networks can be trained effectively without excessively compromising the performance. The ensembles can achieve good learning performance because one of the ensemble's members is able to learn from the correct learning pattern even though the patterns are statistically mixed with erroneous learning patterns.

8. References

[1]S. Mathur" A feature based neural network model for weather forecasting" *World academy of science, engineering and technology* 34, 2007.

[2]R.A. Pielke, "A comprehensive meterological modeling system RAMS," *Meteorology and Atmosphe-ric Physics, Springer*- Verlag Vol. 49, 69-91p,1992.

[3]B.A. Smith " Improving air temperature prediction with artificial neural networks" *International journal of computational intelligence*, 2007.

[4]J. Gill "Training back propagation neural networks with genetic algorithm for weather forecasting" *IEEE 8th international symposium on intelligent systems and informatics serbia*, 2010.

[5]Senduru Srinivasulu, "Extracting Spatial Semantics in Association Rules for Weather Forecasting Image", *Research Scholar Department of Information Technology, Sathyabama University Chennai, India* IEEE 2010

[6]S.S. Baboo and I.K Shereef. " An efficient weather forecasting system using artificial neural network" *International journal of environmental science and development*, vol. 1, no. 4, 2010.

[7]Y.Wang and S.Banavar "Convective Weather Forecast Accuracy Analysis at center and sector levels", *NASA Ames Research center, Maffett Field, Califonia.*

[8]I.S. Isa "Weather forecasting using photovoltaic system and neural network" *Second international conference on computational intelligence, communication systems and networks*, 2010.

[9]Weather.com,http://www.weather.com

[10]AccuWeather.com,http://www.accuweather.com

[11]Statistica SANN-v9_training_manual.

[12] Carpenter, G. and Grossberg, S. (1998) in Adaptive Resonance Theory (ART), The Handbook of Brain Theory and Neural Networks, (ed. M.A. Arbib), MIT Press, Cambridge, MA, (pp. 79–82).

[13] Ping Chang and Jeng-Shong Shih," The Application of Back Propagation Neural Network of Multi-channel Piezoelectric Quartz Crystal Sensor for Mixed Organic Vapours," Tamkang Journal of Science and Engineering, Vol. 5, No. 4, pp. 209-217 (2002).

[14] S. Anna Durai, and E. Anna Saro," Image Compression with Back- Propagation Neural Network using Cumulative Distribution Function," World Academy of Science, Engineering and Technology 17 2006.

[15] Mark Pethick, Michael Liddle, Paul Werstein, and Zhiyi Huang," Parallelization of a Backpropagation Neural Network on a Cluster Computer,"